

# PATENT ABSTRACTS OF JAPAN

(11) Publication number : **07-114444**  
 (43) Date of publication of application : **02.05.1995**

(51) Int.CI. **G06F 3/06**  
**G06F 12/00**

(21) Application number : **05-260633**  
 (22) Date of filing : **19.10.1993**

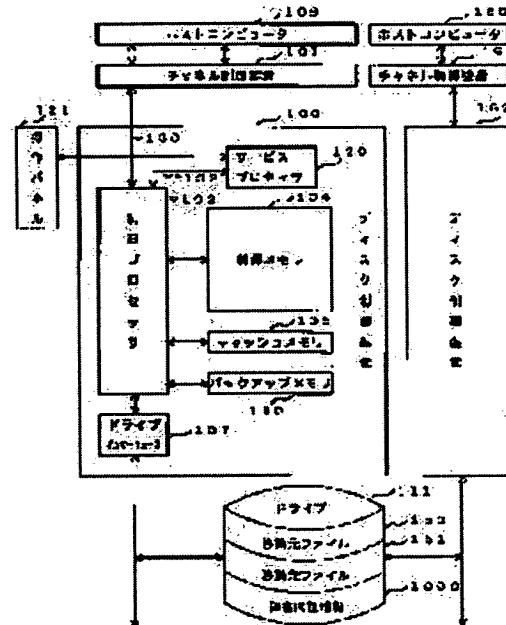
(71) Applicant : **HITACHI LTD**  
 (72) Inventor : **NAKAYAMA SHINICHI**  
**MURATA TOMOHIRO**  
**KURIHARA KENZO**  
**NITSUTA MITSUAKI**

## (54) FILE MOVEMENT CONTROL SYSTEM

### (57) Abstract:

**PURPOSE:** To allow other disk controller to accept a data reference update request from a host computer during file movement or copy processing by storing file processing control information in a drive so as to improve the reliability of the file against a fault of a file processing unit.

**CONSTITUTION:** A disk controller 100 receives a storage location of a movement source file 150 and a movement destination file 151 and a movement command between them from a host computer 109, then allocates a destination track in the unit of tracks, copies data on a source track to the destination track sequentially and connects the source track and the destination track by means of an alternate track pointer. A reference update request from the host computer 109 is executed to data on the source track when the data of the source track are not copied. When the data of the source track are copied, the request is executed to data on the destination track pointed out by the alternate track pointer.



### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-114444

(43)公開日 平成7年(1995)5月2日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 3/06  
12/00

識別記号 庁内整理番号

3 0 4 F  
5 3 1 Z 8944-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 3 O L (全 14 頁)

(21)出願番号

特願平5-260633

(22)出願日

平成5年(1993)10月19日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 中山 信一

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 村田 智洋

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 栗原 謙三

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(74)代理人 弁理士 小川 勝男

最終頁に続く

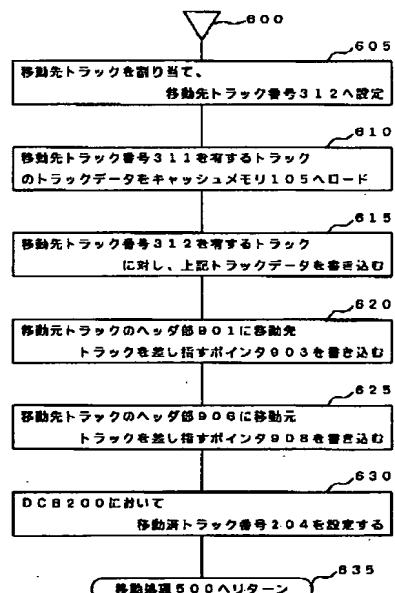
(54)【発明の名称】 ファイル移動制御方式

(57)【要約】

【構成】 ディスク制御装置が、移動元ファイルと移動先ファイルの格納場所の移動指示をホストコンピュータから受け、移動元トラック単位に、移動先トラックを割り当て、移動元トラック上のデータを移動先トラックに順次複写後、移動元トラックと移動先トラックを交替トラックボインタで結ぶ。複写済の移動元トラック上のデータへの参照更新要求は、交替トラックボインタが示す移動先トラックに対して行い、未複写の移動元トラック上のデータへの参照更新要求は、移動元トラックに対して行なう。

【効果】 ディスク制御装置の障害に対するファイルの信頼性が向上し、かつ、ホストコンピュータからのファイル格納場所移動指示により、一つのディスク制御装置が、ファイルの移動、または複写処理を実行している間に移動処理中のファイルへの参照更新要求を他のディスク制御装置のが実行することができる。

図 6



1

## 【特許請求の範囲】

【請求項1】ディスク制御装置と、磁気記憶媒体であるドライブからなるディスクサブシステムにおいて、ホストコンピュータから移動または複写の対象となるファイルの格納場所及び当該ファイルの移動または複写先を含むファイルの移動または複写指示を受けたディスク制御装置が、前記移動または複写の対象となるファイルが格納される移動元トラックまたは複写元トラックに対応してトラック単位に、その移動先トラックまたは複写先トラックを割り当て、前記移動元トラックまたは複写元トラック上のデータを前記移動先トラックまたは複写先トラックに移動または複写した後、前記移動元トラックまたは複写元トラックと前記移動先トラックまたは複写先トラックとの間を交替トラックポインタで結び、複写済みのデータに対する参照更新を前記交替トラックポインタが示す前記移動先トラックまたは複写先トラック上のデータに対して行ない、未複写のデータに対する参照更新を前記移動元トラックまたは複写元トラック上のデータに対して行なうことによりホストコンピュータから前記移動または複写指示の対象ファイルへの参照更新要求を実行することを特徴とするファイル移動制御方式。

【請求項2】前記ディスク制御装置は、少なくとも1トラック分のデータを格納できるメモリを備え、当該メモリを介して、前記移動元トラックまたは複写元トラックから前記移動先トラックまたは複写先トラックへの移動または複写処理を実行することを特徴とする請求項1記載のファイル移動制御方式。

【請求項3】前記ディスク制御装置はさらに不揮発性のバックアップメモリを備え、前記移動または複写処理により前記メモリに格納され、かつ前記移動先トラックまたは複写先トラックへの格納が行なわれていないデータの更新は前記メモリに対して実行し、当該更新データ及び更新対象となったトラックを識別する情報を前記バックアップメモリへ保存し、前記移動先トラックまたは複写先トラックに格納されたデータに対する更新処理は、前記移動先トラックまたは複写先トラックのデータに対して行なうことを特徴とする請求項2記載のファイル移動制御方式。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、ディスク制御装置に関し、特に、同一ドライブ内におけるファイルの格納場所の移動、または複写処理中に、ホストコンピュータからの当該移動対象ファイルへのデータ参照更新要求を実行することにより、ディスクサブシステムのファイル処理のスループットを向上させる方法に関する。

## 【0002】

【従来の技術】従来のディスク制御装置では、例えば、特開平1-230147号公報に開示されているように、コピー元ファイルからコピー先ファイルへコピー処

10 2

理中に、ホストコンピュータからコピー元ファイルへのデータ更新要求があった場合、その更新データをコピー元ファイルへ書き込む。さらに、その更新データアドレスを記憶し、コピー先ファイルへのコピーが済んだことを記憶しておくコピーポインタと更新データアドレスを比較し、そのアドレスがコピー済みであると判定したとき、コピー先ファイルのコピー済み部分に書き込み、未コピーであると判定したは、コピー先ファイルには書き込まないようにして、ファイルコピー処理中にホストコンピュータからのデータ更新要求を実行できる場合があった。

【0003】ただし、上記従来技術では、コピー済み部分を記憶するコピーポインタなど、コピー処理のために使用する制御情報をファイル処理装置内のみに保持して、コピー処理を実現していた。

## 【0004】

【発明が解決しようとする課題】上記従来技術では、ファイル処理中に、コピーポインタなどファイルを処理する際の制御情報（ファイル処理制御情報）を保持したファイル処理装置に電源障害、プロセッサ障害、バス障害などが発生すると、障害発生前に保持していたファイル処理制御情報を喪失してしまうことになり、ファイルの整合性がとれず、ファイルに対する信頼性が低下するという問題があった。

【0005】また、複数のファイル制御装置がファイルを共有する場合、上記従来技術では、複数のファイル処理装置の間にコピーポインタなどのコピー処理のために使用する制御情報を格納する共通制御メモリを持たせなければならなかつた。

【0006】本発明の目的は、ファイル処理制御情報をドライブ内に保持することにより、ファイル処理装置の障害に対するファイルの信頼性を向上させ、ファイルの移動、または複写処理中にホストコンピュータからのデータ参照更新要求を複数のディスク制御装置における任意のディスク制御装置で受け付けることを可能にすることである。

## 【0007】

【課題を解決するための手段】上述した課題を解決するために本発明に係るディスク制御装置は、

（1）ディスク制御装置が、ホストコンピュータから移動元ファイル（複写元ファイル）及び移動先ファイル（複写先ファイル）の格納場所とそれらの間の移動指示を受け、トラック単位に、移動先トラック（複写先トラック）を割り当て、移動元トラック（複写元トラック）上のデータを移動先トラック（複写先トラック）に順次、複写後、当該移動元トラック（複写元トラック）と当該移動先トラック（複写先トラック）を交替トラックポインタで接続する第1の手段。

【0008】（2）当該交替トラックポインタの有無をチェックすることにより、当該交替トラックポインタが

50

存在しなければ、当該移動元トラック（複写元トラック）は、未複写であると判定し、当該交替トラックポインタが存在すれば、複写済みであると判定し、バッファ上に、当該移動元トラック（複写元トラック）が存在するとき、複写処理中であると判定する第2の手段。

【0009】(3) ホストコンピュータからの参照更新要求を当該移動元トラック（複写元トラック）が、未複写であるときには、移動元トラック（複写元トラック）上のデータに対して実行し、複写済みであるときには、交替トラックポインタが指し示す移動先トラック（複写先トラック）上のデータに対して実行し、複写処理中であるときには、キャッシングスロット上のデータに対して実行する第3の手段。

【0010】(4) 移動元ファイル（複写元ファイル）から移動先ファイル（複写先ファイルへ）の複写処理が終了後、当該交替トラックポインタを切断する第4の手段。

【0011】を具備することを特徴としている。

【0012】

【作用】第1の手段で、ファイルの複写を行なうことができる。この間、ホストコンピュータから当該ファイルへ参照更新要求がきた際には、第2の手段で、交替トラックポインタの有無を判定することにより、移動元トラック（複写元トラック）の複写状態を、未複写、複写済、複写処理中のいずれかに定めることができる。その結果により第3の手段で、データの複写状態に合致した参照更新要求を実行することができる。これにより、ファイル移動、または複写処理中にホストコンピュータからの参照更新要求を矛盾なく実行できる。さらに、第4の手段で、交替トラックポインタを切断することにより、ホストコンピュータからのファイル移動、または複写命令を実行完了できる。

【0013】

【実施例】図1は、本発明の一実施例の対象となるキャッシング付きディスクサブシステムの構成図である。

【0014】本発明の前提条件として、ドライブ111は複数のディスク制御装置100、162から共有される。ドライブ111は、ディスク制御装置100により使用されているとき、ディスク制御装置161からのアクセスに対し、ビジー通知を行い、ディスク制御装置162を排他する。あるいは、ディスク制御装置100とディスク制御装置162を排他する情報を持つ。

【0015】図1において、ディスク制御装置100は、上位側でチャネル制御装置101を介してホストコンピュータ109と接続され、また下位側で磁気記憶媒体であるドライブ111と接続される。ドライブ111内には、移動元ファイル150、移動先ファイル151及び障害回復情報1000が格納されている。障害回復情報1000の具体的な説明は、後述する。ディスク制御装置100は、ホストコンピュータ109の要求に応

じてドライブ111にデータのリード、ライトを行なう。ドライブ111とホストコンピュータ109との間のデータ転送を制御するのはディスク制御装置100に内蔵された制御プロセッサ102である。制御プロセッサ102は、チャネル制御装置101、ドライブインターフェース107を介し、ホストコンピュータ109、およびドライブ111と接続する。制御メモリ104は、全ての制御プロセッサ102からアクセス可能な共通メモリであり、ディスク制御装置100がドライブ111をアクセスするための共通制御情報を格納してある。共通制御情報については後述する。キャッシングメモリ105も全ての制御プロセッサ102からアクセス可能なメモリであり、ドライブ111から制御プロセッサ102が読みだしたデータを一時的に格納するために用いられる。制御プロセッサ102は、またサービスプロセッサ120と接続される。操作パネル121からサービスプロセッサ120に対し共通メモリ104内の共通制御情報の更新を指示すると、サービスプロセッサ120が、制御プロセッサ102へ更新要求を送り、選択された方の制御プロセッサ102が制御メモリ104内の共通制御情報の更新を行なう。

【0016】ディスク制御装置162は、上位側でホストコンピュータ160と接続され、チャネル制御装置161と接続され、構成はディスク制御装置100と同じである。

【0017】次に、共通制御情報について説明する。共通制御情報には、ドライブ制御ブロックとファイル移動要求テーブルがある。以下、順に説明する。

【0018】図2は、ドライブ制御ブロック200(Device Control Block: 以下ではDCBと呼ぶ)を示す。DCB200には、7つのデータが格納されている。各ドライブをディスク制御装置100が識別するためのドライブ番号201(Device Connection Address: 以下DCAと呼ぶ)、当該ドライブのトラック長などのドライブ情報を示すドライブタイプ202、移動対象ファイルの移動先終了トラック番号203、移動済トラック番号204、移動元トラックポインタ切断済トラック番号205、当該ドライブへのホスト参照更新要求フラグ206、及びドライブロックフラグ207が格納される。

40 ホスト参照更新要求フラグ206は、ホストコンピュータ109からファイル移動処理中のドライブ111へ参照更新要求があったときに、制御プロセッサ102が、'on'に設定する。参照更新要求が、無いときには、'off'に設定しておく。また、ドライブロック情報207は、制御プロセッサ102が、他の制御プロセッサからのドライブアクセスを排他制御するときに使用し、ロック取得中は'off'に設定し、ロック開放時(アンロック時)には、'off'に設定する。

【0019】図3は、ファイル移動制御情報を示す。本制御情報には、ファイル移動要求テーブル300と移動

処理用ワークエリア310を格納する。ファイル移動要求テーブル300には、移動対象ファイルを有するドライブのドライブ番号301、移動対象となるファイルの格納位置を示す移動元開始トラック番号302、及び移動元終了トラック番号303、移動対象ファイルの移動先を示す移動先開始トラック番号304が格納される。移動処理用ワークエリア310は、トラックの移動処理に使用するエリアで、移動元トラック番号311、移動先トラック番号312を格納する。

【0020】次に、本実施例におけるディスク制御装置100の制御プロセッサ102の動作を説明する。

【0021】図4は、ファイル格納場所移動処理400のメインフローである。ファイル格納場所移動処理400は、ディスク制御装置100がホストコンピュータ109からのファイル移動命令を受領したとき、または操作パネル121を通して、サービスプロセッサ120からのファイル移動命令を受領後、起動される。

【0022】まず、ファイル移動要求テーブル300を読みだす(410)。具体的には、移動対象ファイルを有するドライブ番号301、移動元開始トラック番号302、移動元終了トラック番号303、移動先開始トラック番号304を読みだす(410)。続いて、ドライブのロック情報207を判定し(415)、それが既に‘on’であれば、ホストコンピュータへドライブビジーを返し(425)、その後、処理を終了する(440)。もし、ロック情報207が‘off’であれば、ホストコンピュータへファイル移動命令受諾を報告し(420)、移動処理500を行なう。続いて、移動処理後の後処理800を行ない、ホストコンピュータ109へファイルの移動が完了したことを報告し(430)、処理を終了する(440)。

【0023】図5は、図4における移動処理500のフローである。

【0024】まず、移動元開始トラック番号302を移動元トラック番号311へ設定する(505)。次に、移動済トラック番号204をゼロクリアし(510)、移動対象ファイルを有するドライブ番号301を有するドライブのロックを取得する(515)。続いて、移動元トラックデータを移動先トラック上への複写処理600を実行して、移動対象ファイルを有するドライブのロックを解放する(520)。次に、ホストコンピュータ109からの当該ドライブへの参照更新要求があるかどうかをホスト参照更新要求フラグ206をチェックする(525)。もし、要求フラグ206が‘on’であれば、参照更新処理700を実行する。要求フラグ206が‘off’であれば、移動処理が終了したかどうか判定する(530)。ステップ530では、移動済トラック番号204と移動元終了トラック番号303を比較し、これらが等しければ、移動処理が終了したと判定する。ステップ530において移動処理終了と判定される

と、移動対象ドライブのDCB200に、移動先終了トラック番号203を設定し(540)、移動処理500を終了して、ファイル格納場所移動処理400へリターンする(545)。移動処理が終了していなければ、移動元トラック番号311をインクリメントし(535)、ステップ515に戻って移動処理を繰り返す。

【0025】図6は、図5における移動元トラックデータの移動先トラック上への複写処理600のフローである。

【0026】複写処理600の説明に先立って、図9を用いて本実施例におけるトラックフォーマットについて説明する。図9において、900は、移動元トラックを示し、905は、移動先トラックを示す。901、906は、各々のトラックのヘッダ部を示し、903、908は、交換トラックポインタを示す。また、902、907は、各々のトラックのデータ部を示す。

【0027】まず、移動元トラックの移動先として、移動先トラックを割り当て、移動先トラック番号312へ割り当てたトラック番号を設定する(605)。この時、移動先開始トラック番号304を有するトラックから、割り当てを開始する。次に、移動元トラック番号311を有するトラックからデータ902をキャッシュメモリ105にロードする(610)。移動先トラック番号312を有するトラックに対し、キャッシュメモリ105にロードしたデータを書き込む(615)。移動元トラックのヘッダ部901に移動先トラックを指すポインタ903を書き込む(620)。次に、移動先トラックのヘッダ部906に移動元トラックを指すポインタ908を書き込む(625)。移動対象ドライブのDCB200に移動済トラック番号204を設定し(630)、移動処理500へリターンする(635)。

【0028】図7は、図5におけるファイル移動処理中に、ホストコンピュータ109からの参照更新要求を実行するための参照更新処理700のフローを示す。

【0029】まず、ドライブのロックを取得する(705)。ドライブ111上の参照更新要求トラック(移動元トラック)のヘッダ部901のポインタ903の有無をチェックする(710)。移動先トラックポインタ903があれば、参照更新要求を移動先トラック上のデータに対して実行し(720)、ポインタ903が無ければ、参照更新要求トラック(移動元トラック)上のデータに対して実行する(725)。このように移動元トラックが複写済であるか否かを判定するには、移動元トラックのヘッダ部901のポインタ903の有無をチェックすればよい。続いて、ドライブのロックを開放し(730)、ドライブへのホスト参照更新要求フラグ206を‘off’とし(735)、移動処理500へリターンする(740)。移動対象ファイルを有するドライブのロックを1トラック単位に、取得、解放することにより、ホストコンピュータ109からファイルへの参照更

新要求を受け入れることができる。

【0030】図8は、図4における移動処理後の後処理フロー800を示す。

【0031】処理800は、移動終了した後、移動元トラックと移動先トラック間の交替トラックポインタ903、908を切断する処理である。

【0032】まず、移動元トラックポインタ切断済トラック番号205をゼロクリアし(805)、移動対象ファイルを有するドライブのロックを取得した後(810)、移動先トラック番号312を移動先開始トラック番号304を設定する(815)。次に、移動先トラック番号312を有するトラックのヘッダ部906の移動元トラックポインタ908をゼロクリアする(820)。続いて、移動元トラック番号312を有するトラックのヘッダ部901の移動先トラックポインタ903をゼロクリアする(825)。これにより、移動元トラックと移動先トラック間の交替トラックポインタ903、908が切り離され、移動元トラックの移動が完了したことになる。次に、移動対象ドライブのDCB200に、移動元トラックポインタ切断済トラック番号205を設定し(830)、移動元トラックポインタ切断済トラック番号205と移動先終了トラック番号203を比較する(835)。比較の結果が等しければ、移動対象ファイルを有するドライブのロックを解放し(845)、ファイル格納場所移動処理400へリターンする(850)。また、比較の結果が等しくなければ、移動先トラック番号312をインクリメントし(840)、交替トラックポインタ903、908の切断が終了するまで、交替トラックポインタ切断処理(820、825)を繰り返す。

【0033】処理800実行中は、ドライブのロックを取得しているため、ホストコンピュータ109からの要求に対しドライブビジーとなり、ホストコンピュータ109からの参照更新要求に対する待ち時間が生じる。しかし、この待ち時間は、従来のホストコンピュータ109主導のファイル移動処理のそれと比べて短い。

【0034】図10は、障害回復情報1000の構成を示す図である。

【0035】上述したファイル移動処理は、移動元トラックのヘッダ部901または、移動先トラックのヘッダ部906をディスク制御装置100の責任で更新する。従って、ディスク制御装置100の障害(電源障害など)に備えて、障害回復情報1000を保持しておかなければならぬ。

【0036】図10に、障害回復情報1000の内容を示す。障害回復情報1000は、移動元トラック番号311、移動先トラック番号312、各処理フローの実行ステップ番号1003、移動元トラックのヘッダ部1004(図9の901に対応)、及び移動先トラックのヘッダ部1005(図9の906に対応)を保持してい

る。これらは、ファイル移動処理中に、必要に応じて、移動対象ドライブの障害回復情報1000へ書き込まれる。もし、ファイル移動処理中に、制御プロセッサ102または、103が障害を起し、移動元トラックのヘッダ部901および、移動先トラックのヘッダ部906を失ってもドライブ111に保持してある障害回復情報1000を読みだすことにより、障害発生時のトラックのヘッダ部901、906を知ることができるので、中断した処理を再開することができ、信頼性が向上する。

【0037】本実施例では、ドライブ111から制御プロセッサ102が読みだしたトラックデータを一時的に格納するために、キャッシングメモリ105を用いたが、キャッシングメモリ105の代わりに、1トラック分のトラックデータを格納できるバッファを用いても同様にファイル移動処理を実現することができる。

【0038】次に、第2の実施例として、キャッシングメモリ105とバックアップメモリ110を使用した例について図11を用いて説明する。第1の実施例では、トラック上の更新データをキャッシングメモリ105内に格納しているが、キャッシングメモリ105は、揮発性であるので、電源障害などが発生すると、データを喪失する。第2の実施例では、これに対処するために、キャッシングメモリ105をバックアップする目的で不揮発性のバックアップメモリ110を使用し、更新データを格納する。

【0039】図11は、本実施例におけるファイル移動処理1100のフローである。本処理は、ファイル格納場所移動処理400における移動処理500に相当する。

【0040】まず、移動対象ファイルを少なくとも1つのトラックを含む複数のトラック単位に分割し、キャッシングメモリ105へロードするため、移動対象ドライブのロックを取得する(1105)。次に、複数トラック単位に分割した移動元トラックを1トラック単位ごとにキャッシングメモリ105へロードする(1110)。この時、キャッシングメモリ105上の1トラックのデータを格納する場所をキャッシングスロットと呼ぶ。その後、移動対象ドライブのロックを解放する(1115)。トラック単位内の移動元トラックを全てキャッシングメモリ105へロードしたか否か判定し(1120)、移動しないければ次の移動元トラックをキャッシングメモリ105へロードし、移動済みであれば、以下の処理を行なう。この間、キャッシングメモリ105上へロードされた移動元トラックへの参照更新要求は、キャッシングメモリ105上で実行し、バックアップメモリ110へ更新データを格納し、キャッシングメモリ105上にないトラックへの参照更新要求は、ドライブ111上のトラックで実行する。キャッシングメモリ105上の(ホストコンピュータ109から更新要求があれば、更新されている)移動元トラックのデータをドライブ111の移動先へ複

写するために、移動対象ドライブのロックを取得し(1 1 2 5)、キャッシュメモリ 1 0 5 とバックアップメモリ 1 1 0 を用いたトラック複写処理 1 2 0 0 を実行する。その後、移動対象ドライブのロックを解放する(1 1 3 0)。移動済みの移動元トラックに対応するキャッシュメモリ 1 0 5 上のキャッシュスロットを解放することにより、未複写の複数トラック単位内のデータへの参照更新要求は、ドライブ 1 1 1 上の移動元トラックで実行し、複写済みの複数トラック単位内のデータへの参照更新要求は、ドライブ 1 1 1 上の交替トラックポインタが指し示す移動先トラックで実行する。キャッシュスロットを解放していないキャッシュメモリ 1 0 5 にある移動元トラックへの参照更新要求は、キャッシュメモリ 1 0 5 上のキャッシュスロットで実行し、バックアップメモリ 1 1 0 へ更新データを格納する。次に、キャッシュメモリ 1 0 5 にまだ、移動処理をしていない移動元トラックがあるか否か判定し(1 1 3 5)、あれば、次の移動元トラックを移動処理する。なければ、移動対象ファイルを全て移動先ファイルへ複写したか否か判定し(1 1 4 0)、未複写であれば、移動対象ファイルの移動していないトラックを複写処理する。複写済みであれば、ファイル格納場所移動処理 4 0 0 ヘリターンし(1 1 4 5)、交替トラックポインタ 9 0 3、9 0 8 を切断する。移動処理後の後処理 8 0 0 を実行し、移動対象ファイルが移動を完了したことをホストコンピュータ 1 0 9 へ報告し(4 3 0)、処理を終了する。

【0 0 4 1】図 1 2 は、図 1 1 におけるキャッシュメモリとバックアップメモリを用いたトラックの複写処理 1 2 0 0 のフローである。

【0 0 4 2】まず、キャッシュメモリ 1 0 5 上の移動元トラックのデータが格納されているキャッシュスロットのロックを取得する(1 2 0 5)。移動元トラックデータをトラック単位にドライブ 1 1 1 上に移動先トラックを割り当て、移動元トラック上のデータを移動先トラックへ書き込み、交替トラックポインタで接続する(1 2 1 0)。次に、移動元トラックデータを格納していたキャッシュメモリ 1 0 5 のスロットを解放し(1 2 1 5)、移動元トラックのキャッシュスロットをキャッシュメモリ 1 0 5 上から解放する(1 2 2 0)。この時点で、次からの移動元トラックへの参照更新要求は、ドライブ 1 1 1 上で実行される。その後、キャッシュメモリとバックアップメモリを用いたファイル移動処理 1 2 2 0 0 ヘリターンする(1 2 2 5)。

【0 0 4 3】本実施例において障害時の処理は、第 1 の実施例の処理と同様である。

【0 0 4 4】以上説明したキャッシュメモリ 1 0 5 とバックアップメモリ 1 1 0 を使用した第 2 の実施例のように、キャッシュメモリ 1 0 5 とバックアップメモリ 1 1 0 を使用しても、第 1 の実施例で説明したファイル移動処理を同様に実現できる。

【0 0 4 5】以上、2 件の実施例について説明した。第 1 の実施例では、1 トラック分のトラックデータを格納できるキャッシュメモリあるいは、バッファを用意するだけで容易に実現できる。第 2 の実施例の特徴は、第 1 の実施例よりも容量の大きいキャッシュメモリを具備することにより、複数本のトラックをキャッシュメモリ上にロードすることができる。その結果、ホストコンピュータからの参照更新要求をキャッシュメモリ上で実行でき、第 1 の実施例のように参照更新要求をドライブ上で実行する場合よりも、ドライブのロックを取得する時間が短くなり、ロック取得回数も減ることになって、参照更新要求を長時間待たせることはない。また、不揮発性メモリであるバックアップメモリを備えることにより、キャッシュメモリ内のデータを保持することができ、障害に対する信頼性も向上する。さらに、交替トラックポインタを用いて、更新データを順次、ドライブ上のトラックへ書き込むことでキャッシュメモリ上とバックアップメモリ上に更新データを蓄積する必要がない。

【0 0 4 6】なお、以上の実施例では、ディスク制御装置の動作を説明したが、ドライブ内に交替トラックポインタを保持しているため、ディスク制御装置が、ファイル処理中に、ディスク制御装置のデータ更新要求は、交替トラックポインタをたどることにより実行することができる。また実施例では、ファイルの移動処理に関して説明したが、ファイルの複写処理も同様に行うことできる。

【0 0 4 7】以上説明した実施例によれば、ディスク制御装置が、ファイル格納場所の移動指示をホストコンピュータから受け、独自に、ファイル移動処理を実行し、ファイルの移動中でも、ホストコンピュータからファイルへの参照更新要求を実行することができ、ディスクサブシステムのファイル処理のスループットを向上することができる。また、処理中に発生した障害により、処理が中断されても、ドライブ上の障害回復情報から中断していた処理を続行させることができ、トラックのヘッド部の情報を喪失することができなく、信頼性が向上する。さらに、1 トラック単位にドライブのロックの取得と解放を行なうので、ドライブのロックの解放時には、サービスプロセッサからファイルへの参照更新要求のためにドライブのロックを取得することができ、サービスプロセッサからの要求とホストコンピュータ 1 0 9 からの要求を並行に実行できる。さらに、ファイルを移動処理する際、交替トラックポインタなどのファイル移動制御情報をドライブ内に保持するため、ディスク制御装置の障害に対するファイルの信頼性が向上する。また、ディスク制御装置が、ドライブ中のファイルの移動、または複写処理を実行中に、他のディスク制御装置がホストからのデータ参照更新要求を受け付けることができる。

【0 0 4 8】

50 【発明の効果】本発明によれば、交替トラックポインタ

11

などのファイル移動処理制御情報をドライブ内に保持するため、ディスク制御装置の障害に対するファイルの信頼性が向上する。さらに、ホストコンピュータからのファイル格納場所移動指示により、一つのディスク制御装置がファイルの移動処理を実行している間、ホストコンピュータから移動処理中のファイルへの参照更新要求を他のディスク制御装置が実行することができ、ディスクサブシステムのファイル処理のスループットが向上する。

【図面の簡単な説明】

【図1】本発明の一実施例によるディスクサブシステムの構成図である。

【図2】ドライブ制御ブロックの構成図である。

【図3】ファイル移動制御情報の構成図である。

【図4】第1の実施例におけるファイル格納場所移動処理のフローチャートである。

【図5】移動処理のフローチャートである。

【図6】移動元トラックデータを移動先トラックへの複写する処理のフローチャートである。

10

12

【図7】ホストコンピュータからの参照更新要求を実行する処理のフローチャートである。

【図8】移動処理後の後処理のフローチャートである。

【図9】トラックフォーマットを示すフォーマット図である。

【図10】障害回復情報の内容を示す図である。

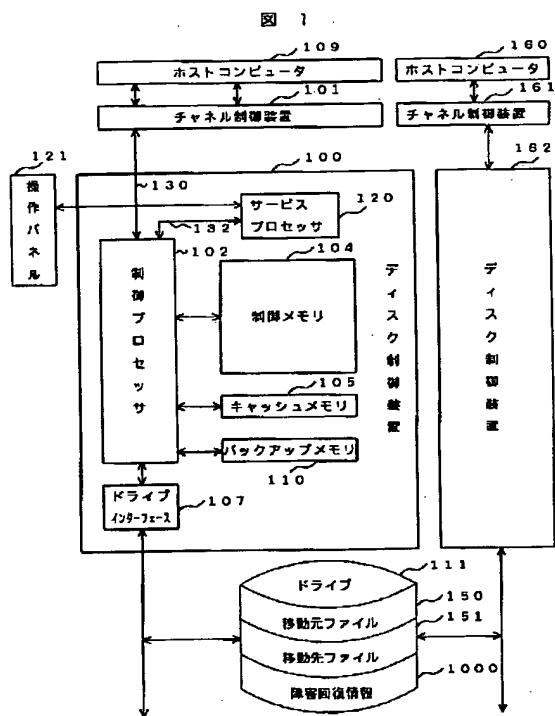
【図11】第2の実施例におけるファイル移動処理のフローチャートである。

【図12】トラックデータの複写処理のフローチャートである。

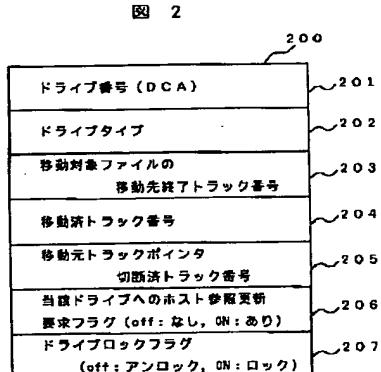
【符号の説明】

100…ディスク制御装置、102、103…制御プロセッサ、104…制御メモリ、105…キャッシュメモリ、107、108…ドライブインターフェース、109…ホストコンピュータ、110…バックアップメモリ、111…ドライブ、120…サービスプロセッサ、130、131…チャネル制御装置と制御プロセッサ間の信号線、132、133…サービスプロセッサと制御プロセッサ間の信号線、150…移動元ファイル、151…移動先ファイル、1000…障害回復情報。

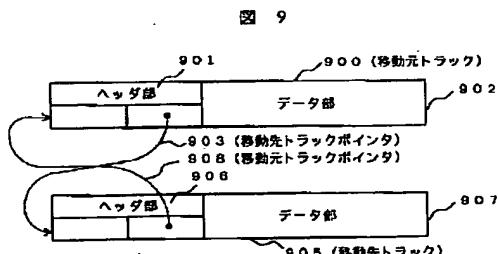
【図1】



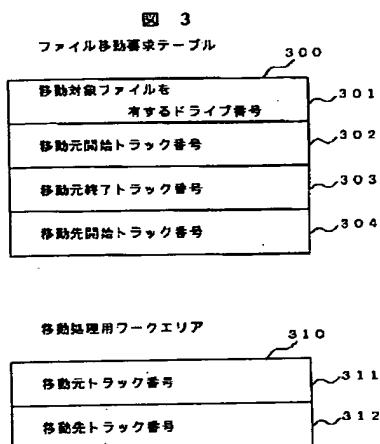
【図2】



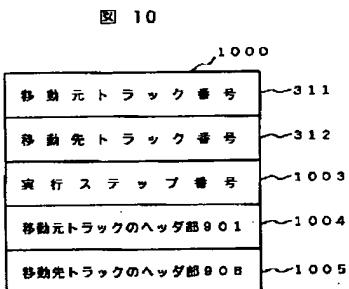
【図9】



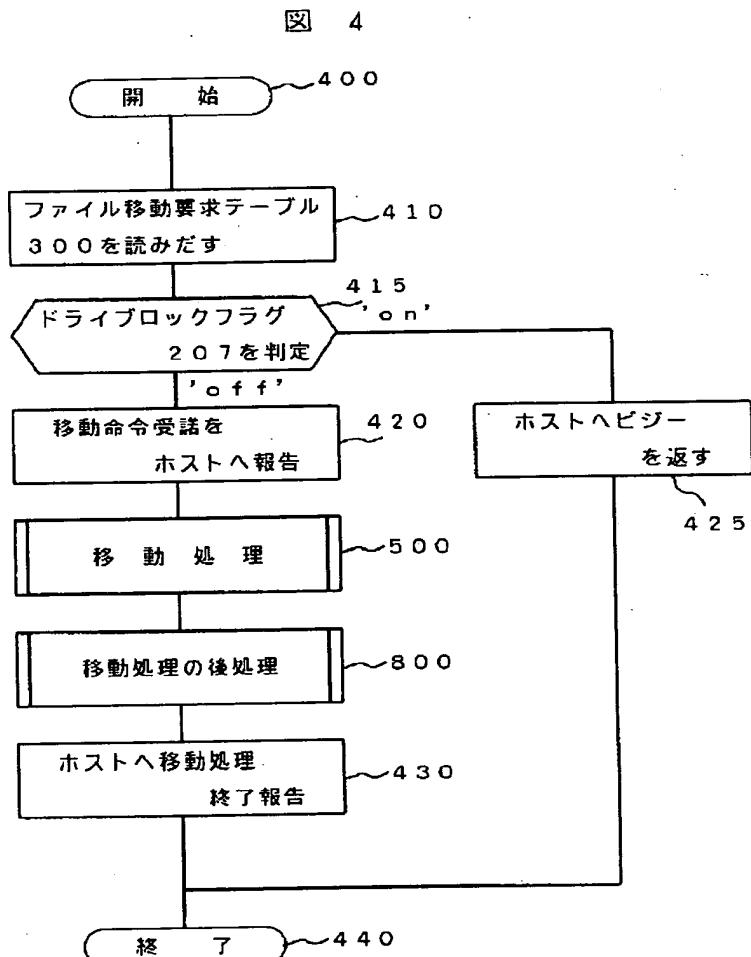
【図3】



【図10】

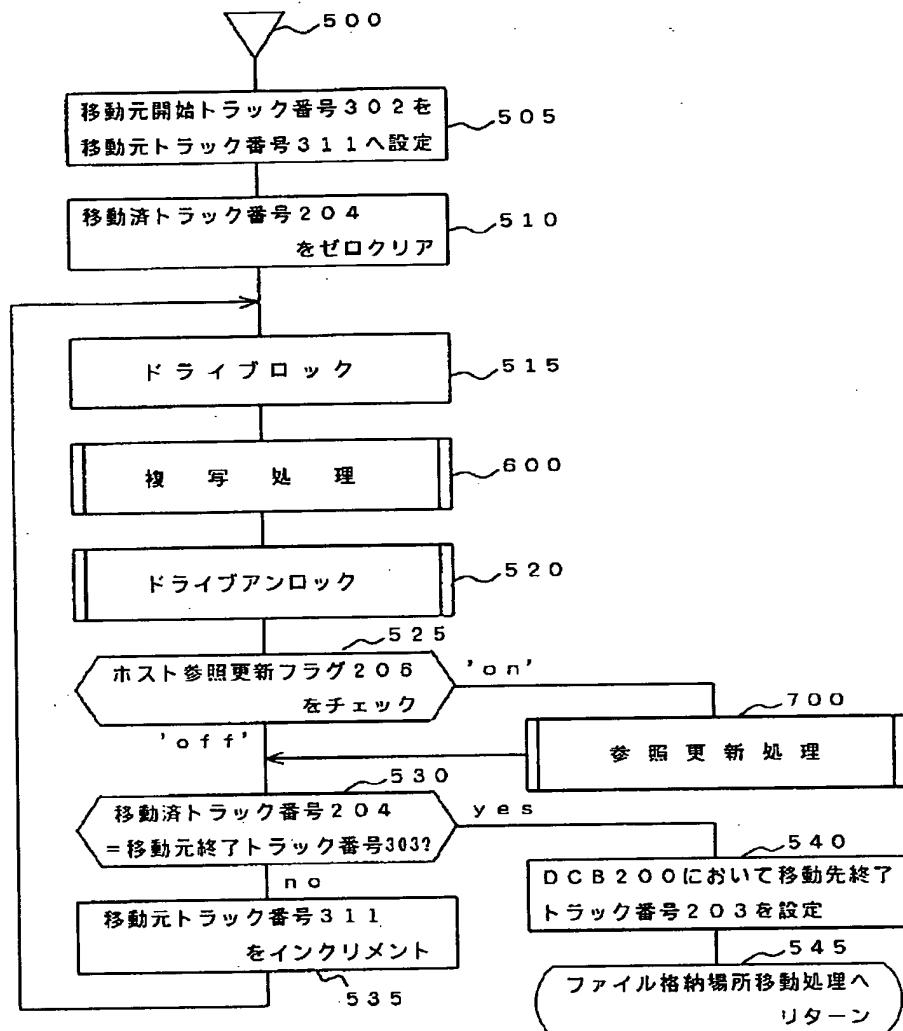


【図4】



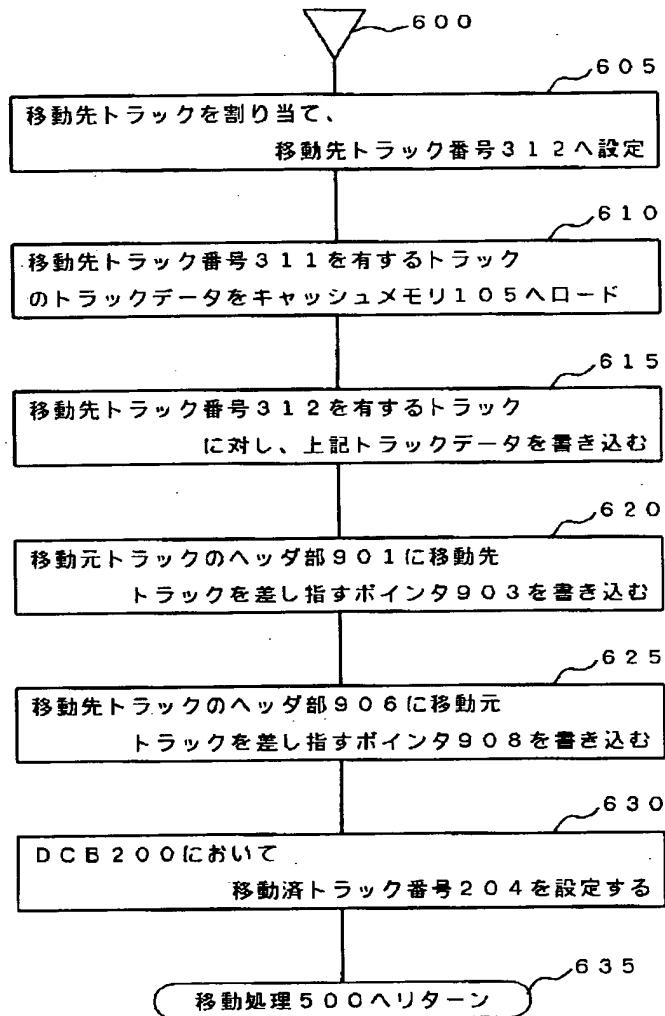
【図5】

図 5



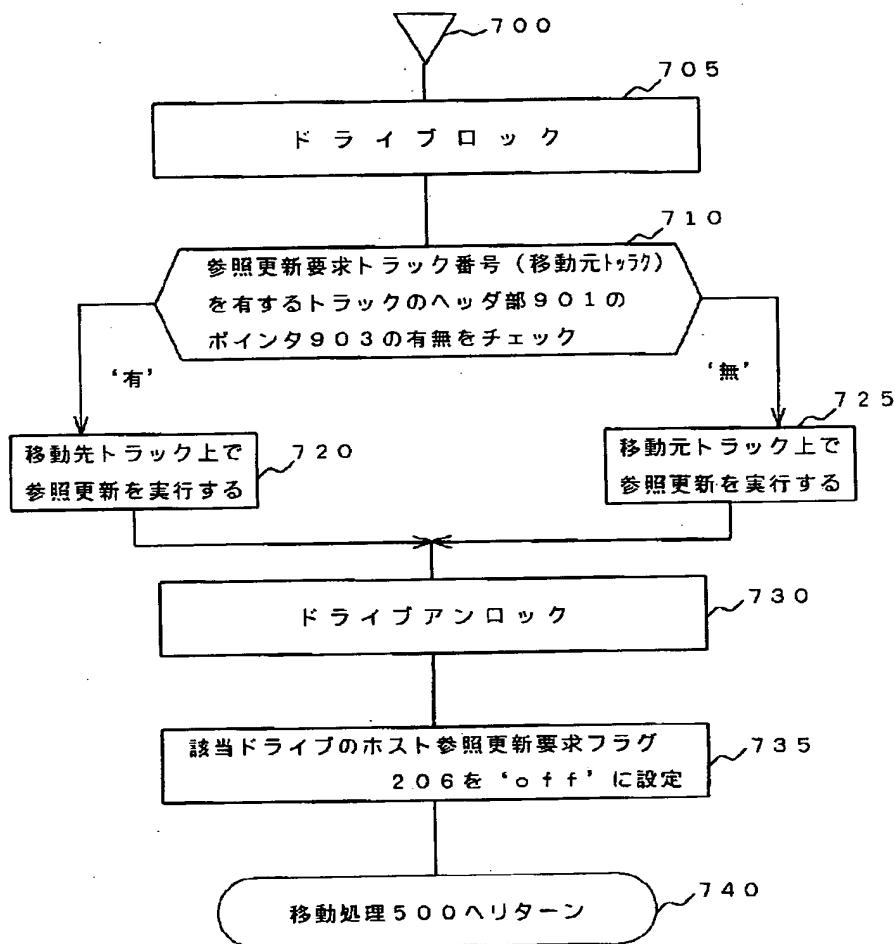
【図6】

図 6



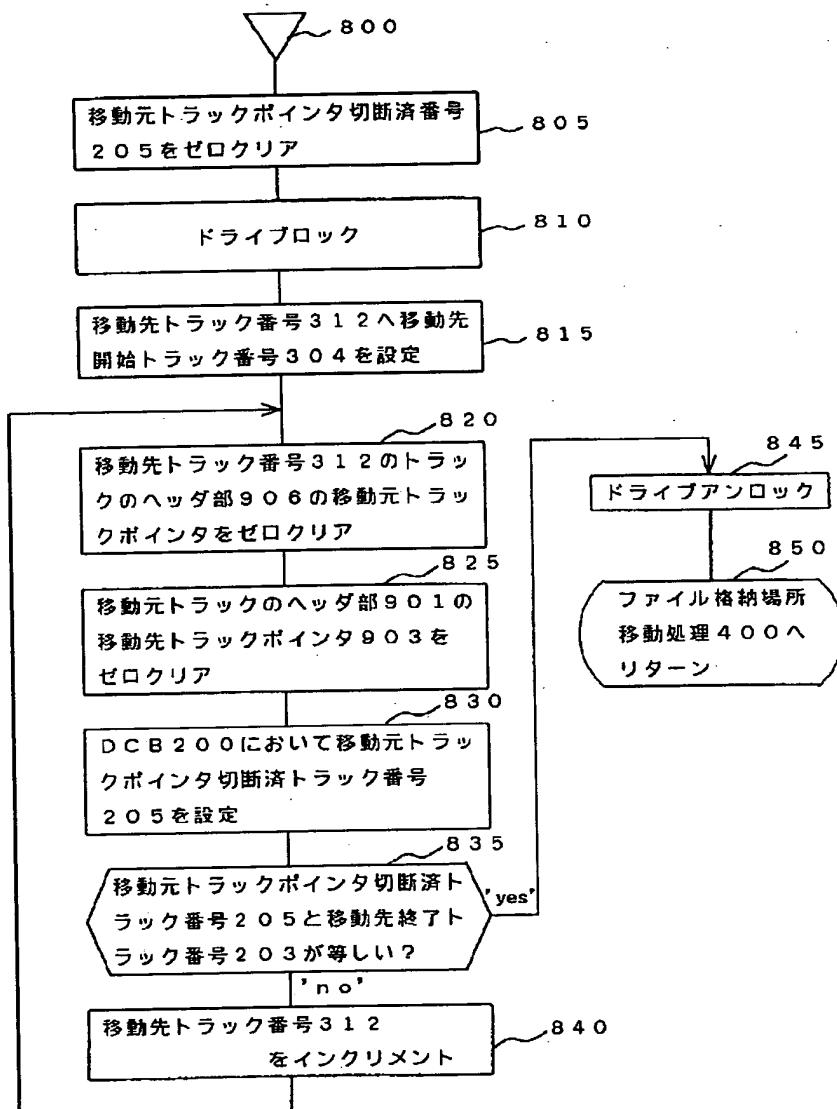
【図7】

図 7



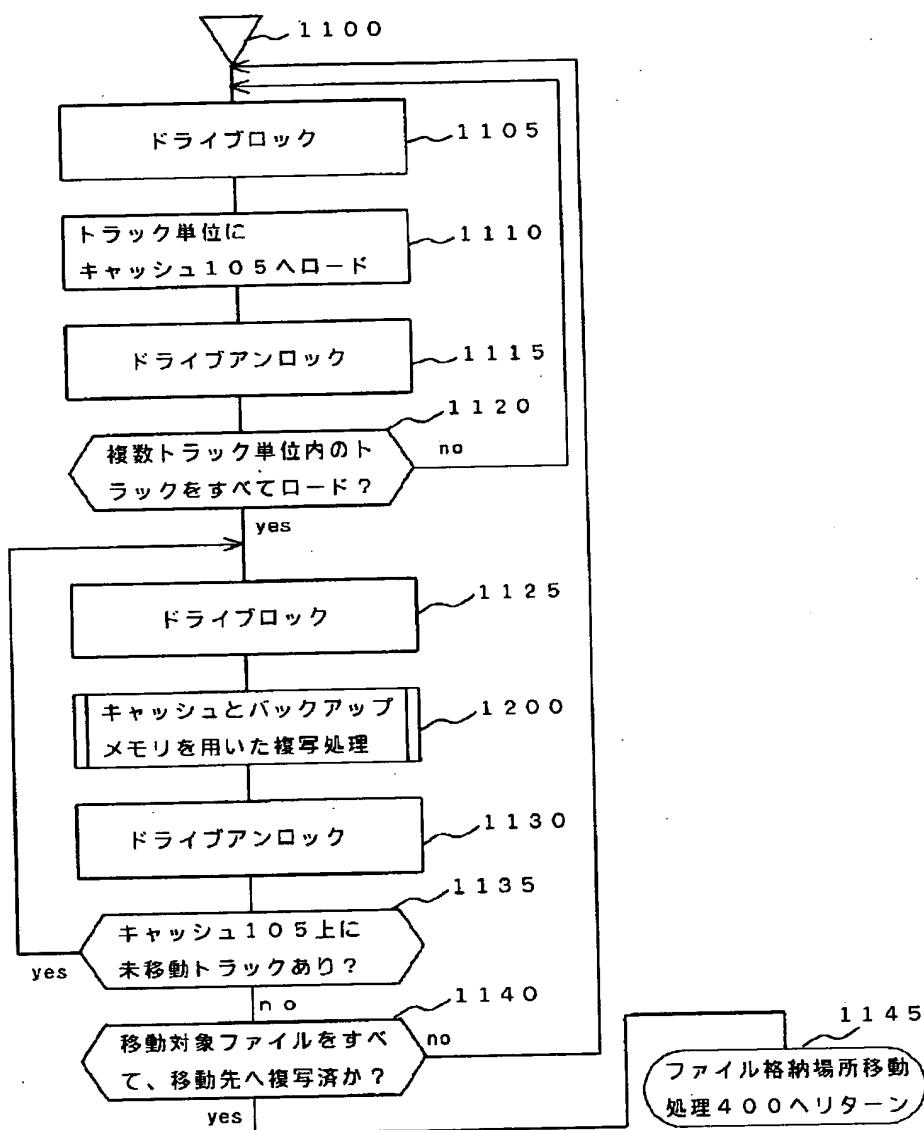
【図8】

図 8



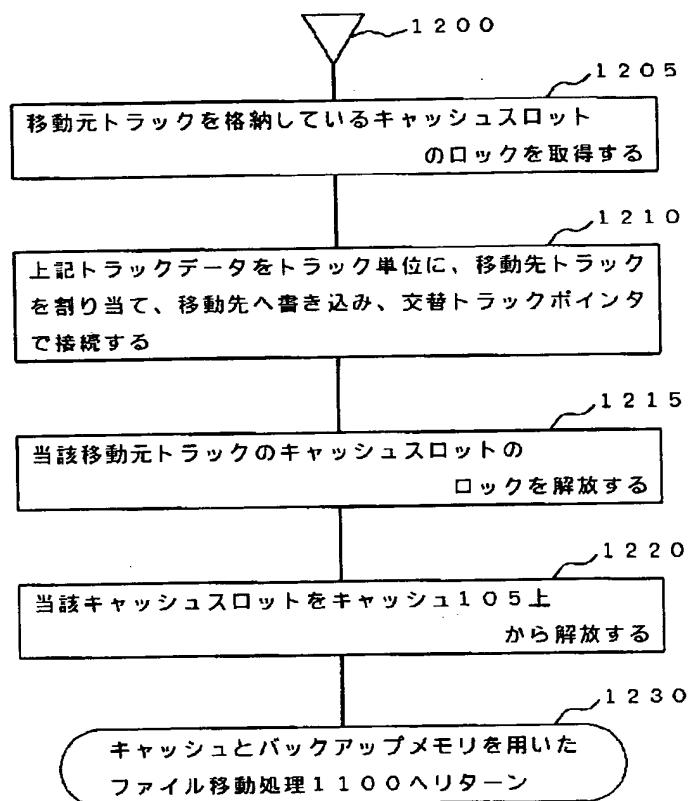
【図11】

図 11



【図12】

図 12



フロントページの続き

(72)発明者 仁田 满秋  
 神奈川県小田原市国府津2880番地 株式会  
 社日立製作所ストレージシステム事業部内